

MLConfig User Manual

Revision 01 JJM 06/02/2009

MLConfig is used to program the Modbus channels in an SHM ML32 datalogger.

1. Installation

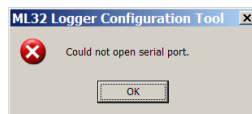
1.1 Software

MLConfig is distributed on the SHM Logger Utilities CD. It does not have a special installation program. Simply copy all the files in the CD directory **MLConf** to a suitable location on the PC. Normally this would be :-

C:\Program Files\SHM\MLConf

Create a shortcut on the Desktop to the executable **MLConfig.exe**.

If you run the application at this stage you may get an error warning :-



1.2 Hardware

To use MLConfig requires a special programming adapter, obtainable from SHM. This is a USB to RS485 serial interface adapter with a special cable to connect it to the programming plug inside the ML32 logger.



A driver installation CD is provided with the programming adapter.

Do not plug the adapter into a USB port on the PC until the drivers have been installed.

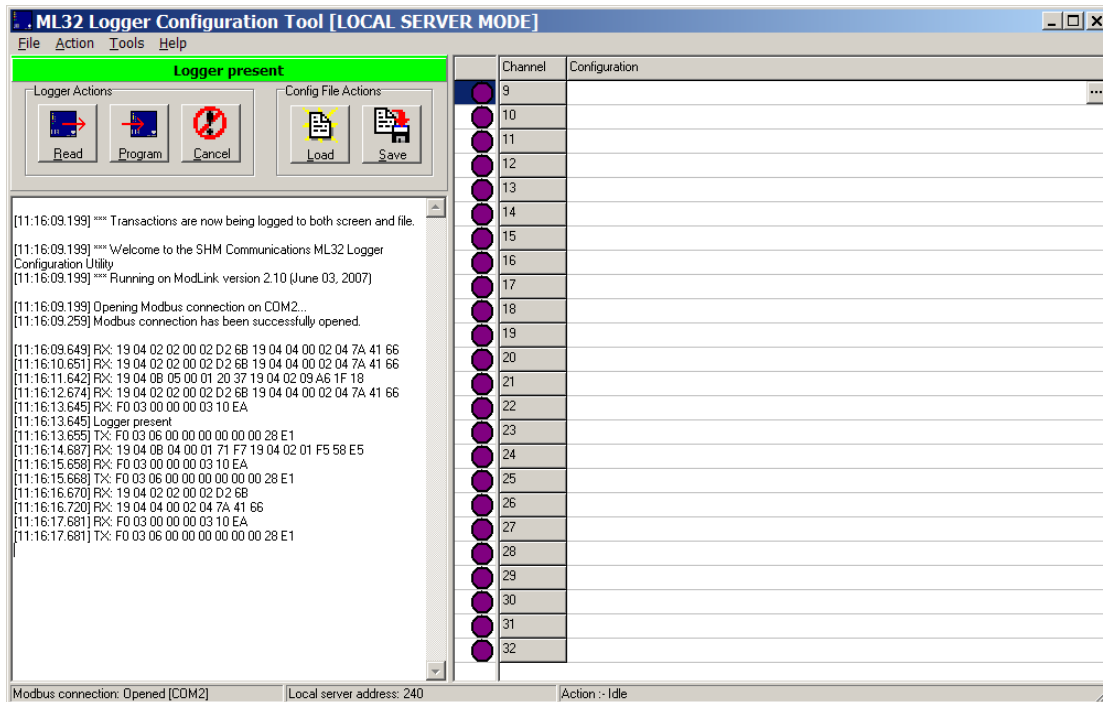
Insert the driver CD into the PC and follow the on-screen instructions to install the driver.

Once the drivers have been installed, plug the adapter into a USB port on the PC, and it should be detected by the operating system in the usual way.

You will need to determine the COM port number assigned to the programmer using Device Manager, and looking under Ports (COM & LPT). Device Manager can be accessed from the Windows control panel, or by using the following command :-

[Start] | Run | devmgmt.msc

2. Main Window



The picture above shows the main window of the application after it has just started up.

The main window is divided into is divided into three regions :-

- The Toolbar. This contains buttons for commonly used actions, and also the "Logger Present" indicator.
- The Diagnostic Pane. This is on the left-hand side and shows communication and diagnostic messages.
- The Channel Configuration Pane. This is on the right-hand side and shows the configuration parameters for each of the logger's Modbus channels (channels 9 to 32).

Each Modbus channel is shown with a traffic light symbol next to it, indicating its status as follows :-

Purple Unknown. The channel configuration has not been read yet.

Green OK. The current logger channel configuration is being shown. Ie. it has been read, or written and read back.

Yellow Modified. The channel configuration has been changed (or a configuration file has been loaded) and is different to the logger configuration. The new configuration needs to be written to the logger.

Red Error. A communication error occurred while reading or writing the logger channel configuration.

3. Configuring the ML32 Logger

Configuring an ML32 logger consists of setting up the programmer, reading the existing configuration, modifying it and writing the new configuration back to the logger.

You can also save the configuration to a file, and load a configuration from file.

3.1 Setting up the Programmer

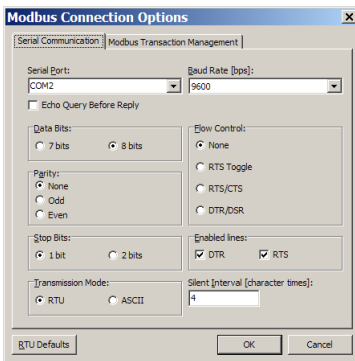
With the software and hardware installed as described in section [1], the ML32 should be powered up and the Programming Adapter cable plugged into the 5-way programming plug on the upper PCB of the logger.

The MLConfig application should be started up. Do not worry if you get the message "Could not Open Serial Port" at this point.

From the menu, select :-

Tools | Modbus Connection Options...

In the following dialog box, set the Serial Port to the one assigned to your adapter.



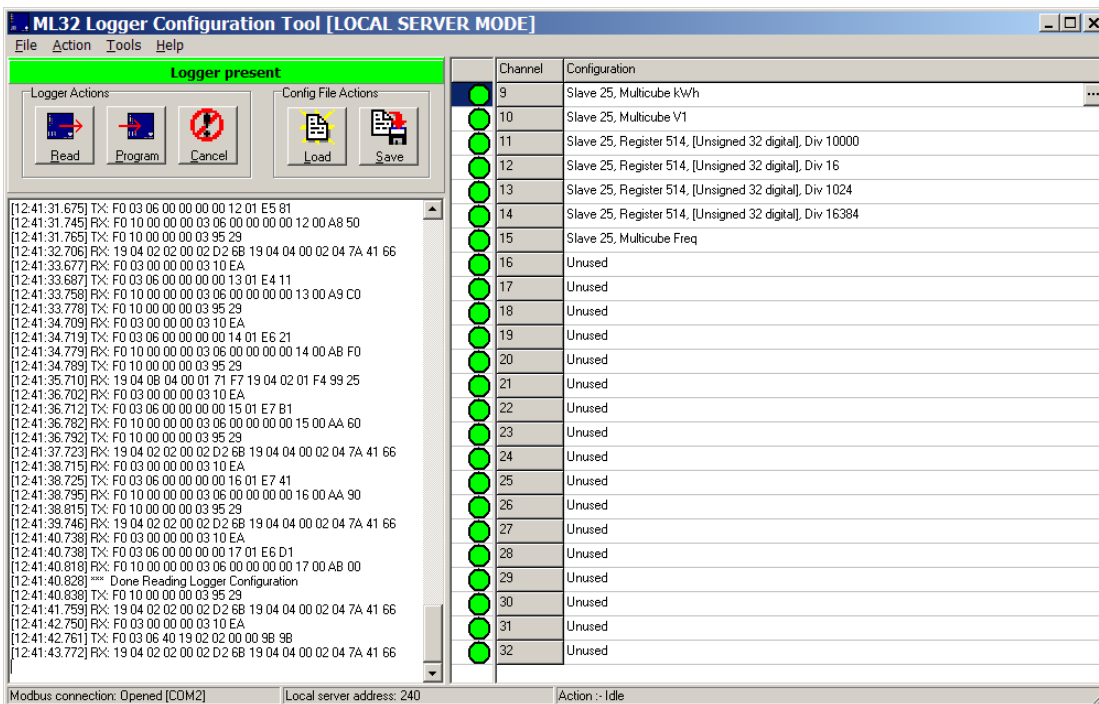
Do not change any of the other settings from the values shown.

Once the serial port is set correctly, you should see communications messages in the Diagnostics Pane, and within 30 seconds, the Toolbar indicator should show "Logger Present".

3.2 Reading the Configuration

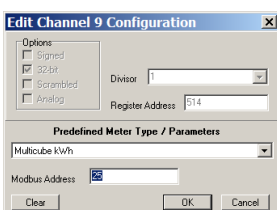
Click the Toolbar button marked [Read] to start reading the current ML32 configuration.

The Traffic Lights in the Channel Configuration Pane should change to green as the channel configurations are read. It takes about 2 seconds to read a channel, so it will take just under one minute to read in the complete configuration :-



3.3 Modifying the Configuration

To modify a channel's Modbus configuration, click on the description in the right-hand pane and then click the [...] button that appears on the right (alternatively, double-click the description). The following dialog box will appear :-

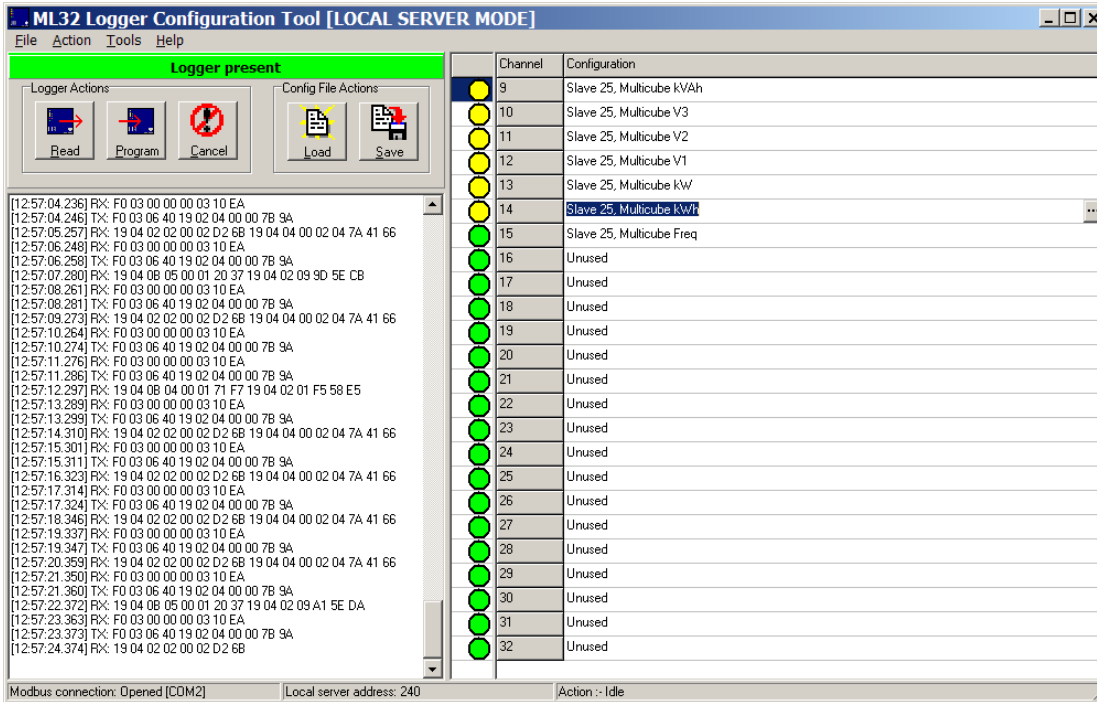


You can change the Modbus Slave Address or select a different pre-defined meter parameter from the list. The upper half of the dialog is normally disabled – to enable entry of these parameters, you must select --Custom-- from the drop-down list. Refer to section [5.2] for setting custom parameters.

To disable the channel, set the Modbus Address to zero. Clicking the [Clear] button will also do this.

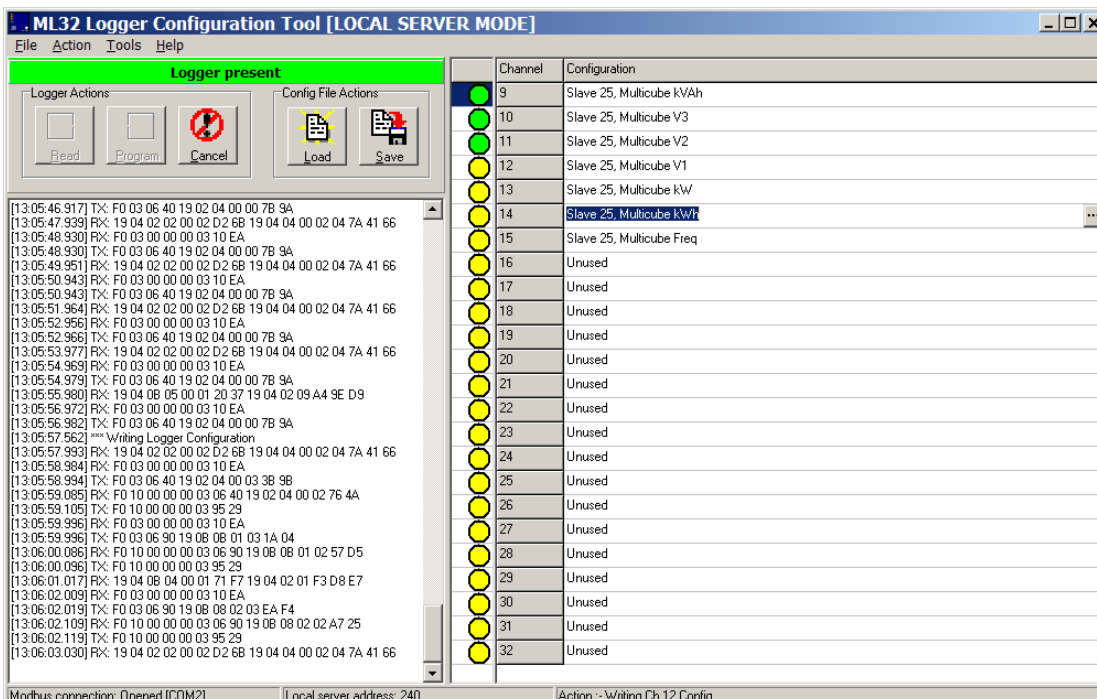
The predefined meter parameters are loaded from a configuration file. SHM can supply a custom configuration file if you are not using one of the standard meter types.

The modified channels will have yellow Traffic Lights beside them, to remind you that these channels need to be programmed into the logger :-



3.4 Programming the Logger

To program the configuration shown on the right-hand pane into the ML32, simply click the [Program] toolbar button. The configuration will be written one channel at a time. The traffic lights will change from yellow to green as the configuration is written and verified :-



3.5 Loading and Saving Configurations

Configurations can be saved to a file and loaded from a file. Loading a configuration from a file will set all the Traffic Lights yellow, to indicate that the configuration shown does not match that of the attached logger.

4. Command Reference

The toolbar button commands are also available from the main menu, which supports some additional commands.

4.1 File Menu

<u>L</u> oad Config	Loads an ML32 channel configuration file.
<u>S</u> ave Config	Saves the ML32 channel configuration shown on the right-hand pane.
<u>E</u> xit	Exits MLConfig.

4.2 Action Menu

<u>R</u> ead Logger	Read and display the ML32 channel configuration. See section [3.2].
<u>P</u> rogram Logger	Write the displayed ML32 channel configuration to the logger. See section [3.4].
<u>C</u> ancel	Cancel the current Read or Program action.

4.3 Tools Menu

<u>M</u> odbus Connection Options...	This allows the COM port used by the programmer to be changed. Do not change any of the other settings.
<u>M</u> odbus Server Options...	This allows the Modbus Slave address used by the programmer to be changed. This should always be set to 240.
<u>L</u> og Transactions To File	This menu item may have a tick beside it. Select the item to turn the tick on and off. If ticked, all diagnostics are logged to a file.

4.4 Help Menu

<u>A</u> bout	This shows version information.
<u>U</u> ser Manual	This will open this User Manual (pdf format - needs Adobe Reader installed on the PC).

5. Advanced Topics

5.1 Modbus Register Addresses

In MLConfig, Modbus Register Addresses are expressed as a physical address, ie. the value transmitted in the Modbus packet, range 0 to 65535.

There is a convention often found in product documentation where the address is expressed as a value in the range 1 to 65536 (one greater than the value transmitted), and is often preceded by a code indicating the Modbus command (3 or 4) used to read the data. This is known as the Modicom Address.

For example if the product documentation states that the kWh register has a Modicom address of 300063, then you would set the Register Address in MLConfig to 62.

The ML32 uses the Modbus command 03h (Read Holding Registers) to read data from slave devices.

5.2 Custom Modbus Configurations

When editing a channel configuration, if you select **--Custom--** as the predefined meter parameter, then you are free to enter custom configuration to allow the ML32 to log data from almost any Modbus RTU device.

You can specify a register address, divisor and options for the channel configuration.

Modbus Slave devices represent their data as an array of 16-bit registers with a Register Address in the range 0 to 65535. Not all register addresses are valid, and the meaning of each register is specific to the device. If more than 16 bits are required to represent an item of data, then it will be spread across several consecutive addresses (eg. 32 bit numbers are represented as two 16-bit registers).

The ML32 supports 16-bit integers (signed and unsigned) and 32-bit integers (signed and unsigned). It does not support floating-point parameters.

There are two ways of representing 32-bit data in Modbus – most-significant word first and least-significant word first. Leave the **Scrambled** box unchecked for most-significant word first and check the box for least-significant word first.

A Divisor can be specified for pre-scaling purposes. Available Divisor values are :-

1, 10, 100, 1000, 10000, 16, 1024, 16384

For example the Total kWh register for a Carlo Gavazzi meter is specified in the documentation as follows :-

Modicom Address	300063
Format	INT32
Scaling	kWh * 10
INT32 Order	LSW->MSW

The corresponding ML32 settings (assuming logging kWh) would be :-

Register Address	62
Divisor	10
Signed	
32-bit	✓
Scrambled	✓
Analog	

5.2.1 Analog vs Digital channels

You must check the **Analog** box for an Analog channel, otherwise it is Digital.

Analog channels have values that can go up as well as down, and must also generate a signed 16-bit result (ie. in the range -32767 to +32767). The divisor is used to ensure that the result fits into this range.

For example, if a meter presents current in the form of millamps as a 32-bit signed integer, then to log the current in Amps you should select a divisor of 1000 (assuming the current does not exceed 32767 Amps).

For Analog channels, you should check the **Signed** box if the value is signed.

Digital channels are typically meter energy registers and always increment. Digital channels are unsigned, and are always logged as 32-bit unsigned integers. If the value to be logged is only 16 bits long, then the ML32 will extend it to 32 bits automatically.

5.2.2 16 and 32 bit Integers

The **32-bit** box should be checked if the parameter is stored as two consecutive 16-bit registers. Otherwise 16 bits is assumed.

If 32-bit, then the **Scrambled** box should also be checked if the two words are stored least-significant word first. This box is not relevant unless **32-bit** is checked.

Check the **Signed** box if the parameter is a signed integer (32-bit or 16-bit). This box is only relevant if **Analog** is checked.

5.3 Configuration Files

MLConfig uses two configuration files, both in the same directory as the executable file :-

MLConfig.ini	This contains all the Modbus configuration information (COM port, timeouts etc).
MeterCfg.ini	This contains all the predefined meter definitions.

Neither of these files should be modified directly by the user. If the Modbus configuration is accidentally changed, you can delete MLConfig.ini, and MLConfig will load default values next time it starts. You should then only need to re-set the COM port (which defaults to COM1).

5.4 Log Files

If the Log Transactions to File option is checked (see section [4.3]) then a log file is generated in a directory off the executable file directory :-

.....\Logs\MLCONFIG.log

5.5 Communications Timeouts

If MLConfig suffers from communications timeouts, which can show up as intermittent indications of **Logger Not Detected** or red Traffic Lights when reading or writing the ML32 configuration, then this can be due to slow operation of the USB serial driver.

You should try closing down all other applications on the PC to make it run faster.